RL-TR-97-161
Final Technical Report
October 1997

SYNCHRONIZATION AND TIMING IN ALL-OPTICAL NETWORKS

MJ Photonics, Inc.

Paul R. Prucnal

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-97-161 has been reviewed and is approved for publication.

APPROVED: David Gruczya

DAVID J. GRUCZA
Project Engineer

FOR THE DIRECTOR: Gary D. Barmore

GARY D. BARMORE, Maj, USAF
Deputy Director
Surveillance & Photonics Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/OCPA, 25 Electronic Pky, Rome, NY 13441-4515. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1997	**FINAL, Jun 96 – Jun 97**

4. TITLE AND SUBTITLE
SYNCHRONIZATION AND TIMING IN ALL-OPTICAL NETWORKS

5. FUNDING NUMBERS

C - F30602-96-C-0174
PE - 62702F
PR - 4600
TA - P5
WU - PP

6. AUTHOR(S)
Dr. Paul R. Prucnal

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MJ Photonics, Inc.
9 Kimberly Court
Princeton NJ 08540

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory/OCPA
25 Electronic Pky
Rome NY 13441-4515

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
RL-TR-97-161

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: David J. Grucza, OCPA, (315) 330-2105

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this report, a platform design for packet-switching optical time division multiplexed (OTDM) networks operating at ultrafast bit-rates is presented. The design is implemented in a prototype 8-node transparent shuffle network. The flow control in this network is based on a self-routing scheme developed to reduce the address processing delay at each node and make the network scalable. Deflection routing is used as the contention resolution principle. Several issues on designing optical networks, such as the node configurations and finding the optimal routing algorithm for maximum throughput, are discussed. Subsystem experiments demonstrate the functionality of the components including a 100 Gbps optical packet compressor and a parallel array of Terahertz Optical Asymmetric Demultiplexers (TOADs) used for header address recognition. Precision delay lines are used for synchronization and timing, based on optical self-clocking techniques. The overall capability of the ultrafast packet-switching testbed is evaluated.

14. SUBJECT TERMS
optical communications, ultrafast optical networks, transparent optical networks, fast packet switching, photonic switching, optical switching, self-routing, deflection routing, optical time division multiple access (OTDM)

15. NUMBER OF PAGES
60

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UNLIMITED**

Table of Contents

1. Introduction 1

2. Network and Node Designs 3

2.1 ShuffleNet 6

2.2 Design of Transparent Optical Network Node 7

2.3 Self-routing and Self-clocking Schemes 9

2.4 Routing Decision Making 12

2.5 Network Throughput 14

3. Subsystem Experiments 15

3.1 Optical Packet Generation and Compression 16

3.2 Packet Header Demultiplexing 18

3.3 Packet Demultiplexing at Receiver 20

3.4 Pointer Control for Self-routing 22

3.5 Controller Electronics 22

3.6 Main Routing Switch 24

4. Network Demonstration 24

5. Conclusions 26

References 28

Table of Figures

Figure 1: (page 32) A 64-node Shuffle network

Figure 2: (page 33) Basic node structure in transparent OTDM networks. Solid lines are optical fibers, dotted lines are electrical switch control lines. The node accepts two incoming packets from other nodes and route outgoing packets to two output ports.

Figure 3: (page 34)Schematic of packet routing switch subsystem showing incoming data from previous nodes, memory, and node receiver.

Figure 4: (page 35) Throughput performance in 8-node and 64-node SNs with (a) blocking main routing switch; (b) strictly non-blocking switch

Figure 5: (page 36) Network delay: the expected number of hops between a random pair of source and destination nodes in 8-node and 64-node SNs with blocking and nonblocking switches

Figure 6: (page 37) Schematic of the optical packet compressor. T: incoming and τ: outgoing bit periods.

Figure 7: (page 38) Illustration of TOAD operation with clock and data at orthogonal polarizations

Figure 8: (page 39) Schematic of header receiver subsystem showing ultrafast serial to parallel conversion

Figure 9: (page 40) Two ultrafast packet receiver designs using TOADs in two different configurations: (a) parallel and (b) serial configurations

Figure 10: (page 41) Implementation of the self-routing scheme using 2 crossbar switches.

Figure 11: (page 42) Schematic of controller board

Figure 12: (page 43) Experimental setup of ShuffleNet node showing packet generation, demultiplexing, and routing subsystems

Figure 13: (page 44)Experimental routing path for test packet in simulated 8-node ShuffleNet

Figure 14: (page 45) Packet format used in the system demonstration. A 16-bit header containing four 4-bit groups for routing and a payload of zeroes was used.

Figure 15: (page 46) Three bits of the demultiplexed address group from the 16-bit header are shown for each node along the routing path: (a) TOAD output displayed on a bandlimited oscilloscope and (b) ECL digital threshold output sent to routing controller.

Figure 16: (page 47)The routing of the optical packet through the 4x4 switch is shown for each node along the routing path: (a) routing switch control signals generated by the controller, and (b) optical output of routing switch.

1 Introduction

The multi-terahertz bandwidth of single-mode optical fiber has provided the main impetus for high bit rate lightwave communications research in the areas of multiprocessor interconnects and optical parallel processor local area networks (LANs). Recent developments and advances in high repetition rate laser sources, optical amplifiers, electro-optic modulators, and novel all optical demultiplexing techniques have demonstrated the feasibility and reliability of many of the subsystems required for multi-gigabit/s data transmission. Transparent optical networks (TONs) can most effectively utilize the inherent speed these optical components have to offer. TONs eliminate electro-optic conversion required for packet regeneration at each node greatly increasing the bandwidth capacity and throughput delay performance of the network. Simplified routing control using only a few bits reduces the complexity of supporting electronic hardware.

Much of the current research work on TONs has focused on wavelength division multiplexing (WDM). Several demonstrations of TONs based upon WDM [1] have been performed including single-hop broadcast and multihop transport network topologies [2, 3]. WDM networks, however, still have several difficult technical challenges to overcome. To fully utilize the bandwidth capacity of optical fiber, many closely spaced wavelength channels must propagate simultaneously on the fiber. It has become difficult to construct rapidly tunable laser sources and receivers (filters) at each node to select the desired channel. The scalability of WDM networks is

determined by several limiting factors including spectral dependent optical amplifier gain, fiber nonlinearities that limit the channel spacing, stimulated Raman Scattering, and four-wave mixing [4]. Furthermore, frequency registration among interconnecting WDM LAN gateways can involve complex decision algorithms, complicated frequency conversion techniques, and may even require time consuming packet regeneration.

We believe high bit rate TONs for multiprocessor interconnects will be best realized with an optical time division multiplexed (OTDM) packet-switched network architecture. To fully utilize the bandwidth of optical fiber, picosecond pulses are spaced closely together ($\sim$10 ps) and a return-to-zero (RZ) modulation format is typically applied. While the total capacity of TDM and WDM networks may essentially be the same, TDM has better throughput delay performance and faster single channel access time for high data rate end users such as HDTV video servers, terabyte media banks, and supercomputers [5]. OTDM techniques can be used to greatly increase the bandwidth of a single wavelength channel [6].

The hardware subsystems and node architectures used to implement high bit rate OTDM networks are typically much simpler than those employed in multichannel WDM networks [5]. This feature coupled with a single-bit self routing strategy, makes the supporting electronics required at each optical node fast, and the network is easily scalable. By using recently developed, ultrafast all optical demultiplexing devices [7, 8], OTDM data rates in excess of 250 Gbps with a bit error rate $\leq 10^{-9}$ are possible such as the Terahertz Optical Asymmetric Demultiplexers [9, 10].

For this report, we have developed a testbed to study both the theoretical aspects and physical implementation issues associated with high bit rate multi-hop, packet-switched OTDM networks. We present a comprehensive analysis of our multi-hop architecture and present experimental results for each of the subsystems used to demonstrate the network. The remainder of the report is organized as follows: in section two, we will discuss the multihop architectures and the application of a special network topology, ShuffleNet, in optical networks, and followed the design of the network node structure with low loss and efficient flow control schemes. Based on our design, network throughput under ideal conditions is analyzed for 8-node and 64-node ShuffleNet. Section three presents the implementation and operation of the necessary components in the network, and the demonstration of packet routing in a simulated 8-node shuffle network.

2 Network and Node Designs

An optical TDM (OTDM) network is either a single-hop or a multihop network. Single-hop networks are generally broadcast networks. A typical example can be one that is simply constructed using a passive star coupler as the shared medium. Each node is equipped with a transmitter and a receiver. At the transmitter, optical data streams are generated via E/O modulation with electronic data, and then time-multiplexed with signals from the other nodes in the network into a single combined data stream. The resulting multiplexed data stream is then broadcasted to

every node by the star. The demultiplexer at the receiver of each node is tuned to its assigned channel and is capable of retrieving the appropriate data from the broadcast stream using rapidly tunable optical delay lines. The allocation of channels can be performed statically at the bit-level or packet-level. Single-hop OTDM networks generally provide larger throughput and shorter delays than multihop networks. However, statistical multiplexing in packet-switched multihop networks leads to much more efficient utilization of network bandwidth under bursty traffic, and the multihop approach is also more practical for geographically distributed networks.

In the class of two-connected multihop networks we investigate, each node is capable of transmitting and receiving packets to and from the network and routing packets in transitions. The nodes are primarily built with photonic space-division $LiNbO_3$ crossbar switches. These switches provide full transparency and scalability in optical networks. With statistical multiplexing, bandwidth in multihop networks is allocated dynamically on demand — available channels are assigned concurrently to those nodes with new packets to transmit. Time-multiplexed packet streams are carried on dedicated optical fibers connecting the nodes on a single wavelength. Each packet has a header containing the necessary routing information and a payload containing the data. The lengths of the packets are fixed and equal and should be chosen based on the transmission bit rate, packetization delays, and routing processing delays at each node in the network. Packet length defined by the Asynchronous Transfer Mode (ATM) standard is 53 bytes, but in ultrafast optical networks operating at

over 100 gigabits/second, significantly longer packet lengths can utilize the enormous bandwidth available in TONs more efficiently. The address information carried in the packet headers can be explicit node addresses or specially coded routing tags. In a self-clocking network, a packet also carries clock pulses for synchronization, both at bit- and packet-level. When a packet arrives at a node in the network, the clock pulses are extracted and the address information is examined by an electronic controller for making an appropriate routing decision. Address coding schemes are often employed so that the routing and flow control in the network can be made as simple as possible to minimize the address processing delay. The controller sets the states of the switches to allow generation and absorption at the node and route packets using shortest-path algorithms. In the cases when there are multiple packets competing for the same output port at the node, the controller resolves contention based on given priority schemes for optimal throughput and delay performance.

Some special network topologies, such as regular two-connected Manhattan street network (MSN) and ShuffleNet (SN), have been studied for their applications to multihop lightwave TDM networks [14, 15]. Both MSN and SN offer a highly desirable self-routing feature, which makes it possible for routing decisions to be made from processing only a small fraction of the information carried in the headers. A comparison of 64-node MSN and SN in [12] under deflecting routing revealed that the throughput of SN is higher than MSN for every offered load at this size. In this paper, ultrafast packet switching is demonstrated experimentally in an 8-node ShuffleNet, and performance simulations are calculated for both 8-node and 64-node shuffle networks.

2.1 ShuffleNet

A ShuffleNet, denoted as $N(q,k)$, has $N = kq^k$ nodes arranged in k columns of q^k nodes each connected in perfect shuffle by unidirectional links, the last column of the network is wrapped around to the first column cylindrically. Each of the p^k users in a column has p arcs directed to p different nodes in the next column. Numbering the nodes in a column from $i = 0$ to $p^k - 1$, node i has arcs to node j, $j+1$, .., and $j+p-1$ in the next column, where $j = (i \bmod p^{k-1})p$. Figure 5 1 shows the configuration of a 64-node ShuffleNet.

A measure of the compactness of multihop networks is the number of nodes that a packet needs to visit taking the shortest path between any pair of nodes. The expected number of node visited between two randomly selected nodes in a SN is given by

$$D = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)(kp^k-1)} \tag{1}$$

The diameter of a SN is given by $d_{max} = 2k - 1$, which is defined as the maximum number of hops a packet needs to transit from the source node to the destination node without deflection. The maximum increase in distance, expressed in number of hops, due to a single deflection is called the deflection cost, which is equal to k in SN. This is a good measure of the network performance degradation under heavy load.

If a packet is at a distance $d > k$ away from the destination node, there are $d - k$ don't care nodes, where the packet can take either one of the two output links to

reach the destination with equal distance. A large number of don't care nodes in the network can keep the occurrence of deflections low under high network load.

2.2 Design of Transparent Optical Network Node

The basic structure for a transparent node in a two-connected multihop network is as shown in Fig. 2. All the nodes in the network are connected by dedicated fiber links. Each node has a main switch which consists of $LiNbO_3$ crossbar switches connecting two input links to two output links and a transmitter and a receiver which are capable of injecting and absorbing packets from both links. When packets arrive at the two inputs, a small fraction of the signal power is tapped off by an optical coupler and sent to the header recognition block, where the bits containing routing information are read by an array of fast demultiplexers and then processed by a high-speed electronic routing controller. Fiber delays are inserted between the couplers and the main switch to allow the controller to set the crossbar switches in advance of packets reaching the main switch.

The preferred output links are determined by the controller to provide the packets with the shortest path to their destinations. If both packets on the two input links wish to take the same output link, a contention occurs. Contentions in optical networks are often resolved by a deflection routing algorithm with limited optical buffering to keep the node structure simple. An optical buffer is implemented by inserting a recirculating fiber loop between a pair of input and output ports. It has been shown that adding a single buffer greatly decreases the probability of deflection,

while adding two or more buffers do not further improve the performance significantly [15].

A general structure of the main routing switch can be represented by a quadruple (S, L, I, O), where S is a set of 2×2 crossbar switches, L is a set of links, I is a set of input ports and O is a set of output ports. The crossbar switches are connected in multistages with links only allowed in adjacent stages. A switch is said to be *blocking* if for some $x \in I \cup S$ and $y \in O \cup S$, a request for connection (x, y) cannot be satisfied. A switch is *strictly nonblocking* if one can choose routes arbitrarily and always be guaranteed that any new connections can be satisfied without rearrangements. A switch is *rearrangeably nonblocking* if for a given set of connections in progress and any pair of idle terminals, the established connections can be reassigned new routes so as to make it possible to connect the idle pair.

It requires a minimum of five 2×2 crossbar switches to construct a rearrangeably nonblocking 4×4 routing switch with $I = \{in\_1, in\_2, mem\_in, TX\}$ and $O = \{mem\_out, out\_1, out\_2, RX\}$, and six for a strictly non-blocking switch [13]. This implies that a nonblocking routing switch will have 3 or more stages. A typical commercially available $LiNbO_3$ switch has a loss of 3.5 dB. Any rearrangeably nonblocking structure with five crossbar switches leads to uneven loss suffered by packets taking different paths through the routing switch, the highest loss being the same as in a strictly nonblocking switch at more than 10 dB. One of the issues we faced in our experiment was to compensate for the high loss in the optical network. Therefore, the

four-switch blocking structure in Fig. 3 was selected for the network demonstration to keep the node loss at a minimum. Every packet flows through two crossbar switches and experiences the same loss, approximately 7 dB. The order of elements in I and O are arranged so that the input ports *in_1* and *in_2* share the same 2×2 crossbar switch, and *mem_in* shares the other with TX. This ensures the fairness among incoming packets arriving at the input ports since the routing algorithm should route the packets in the optical buffer first to minimize delay due to buffering.

2.3 Self-routing and Self-clocking Schemes

Having defined the node structure, this section specifies how routing of packets is handled in the network. Routing decisions at each node can be made by reading either all or partial information in the packet headers. One method to determine an output for each packet at each node is to search in a look-up table which contains all necessary information on network connections to find out the appropriate output port for the packet to reach its destination with the shortest path. In networks with irregular and random topologies, this look-up table method is the only method to use. One problem with this method is that each node needs to keep its own look-up table and continuously update it. More seriously in an ultrafast optical network, the amount of time it takes to search the look-up table contributes significantly to the processing time required to make routing decisions, which ultimately limits the packet rates that can be achieved in the network.

In ShuffleNet, routing decisions can be made based on a single bit information in the header; this is called a *bit-level packet-switching scheme* (BLPS). One well-known scheme is the *destination-tag routing scheme* [16] that uses a binary destination address as a tag for routing a packet. Starting from the source node, the complete path to the destination can be determined by following the tag's binary values [18]. By again numbering the nodes in each column from $i = 0$ to $p^k - 1$, a destination address can be represented as $(d_{k-1}d_{k-2}\cdots d_1 d_0)$. Since the longest path length of shuffle networks without deflection is $2k-1$, $2k-1$ bits in a routing tag are enough for any source and destination pair. At don't care nodes which are more than k hops away from the destination, a packet can take either output ports. The routing tag H can be represented as $(2k-1)$-bit binary numbers as follows:

$$H = (\delta_{2k-1}\delta_{2k-2}\cdots\delta_k d_{k-1}d_{k-2}\cdots d_1 d_0) \tag{2}$$

where $\delta_{2k-1}, \delta_{2k-2}, \cdots, \delta_k$ are *don't care* bits which can be either 0 or 1.

Based on the destination-tag BLPS scheme, we developed a new method that is more suitable for routing at the 4×4 nodes in our network as described in the previous section. Each bit in H is encoded in three bits: a care/don't care (DC) indication bit, and two bits representing an output port. The resulting modified routing tag $\mathcal{H}$ is now

$$\mathcal{H} = \underbrace{[100]\cdots[100]}_{(k-1)\text{ don't care nodes}} \quad \underbrace{[0b_1^k b_0^k][0b_1^{k-1}b_0^{k-1}]\cdots[0b_1^0 b_0^0]}_{k\text{ care nodes (k+1..1st)}} \quad \underbrace{[011]}_{\text{RX port (0th bit-group)}} \quad ; \tag{3}$$

It is noted that in addition to the routing information about care and DC nodes, three more bits are included as a bit-group for routing the packet to RX in the destination node. Thus, the total number of routing bits required in the header is $3(2k-1)+3 = 6k$. Upon arrival at a particular node in the network, only one of the bit-groups in the packet header will be used to make the routing decision.

Each packet carries a clock pulse in a different polarization and is used at each node to synchronize the demultiplexers and routing controller. When a packet is generated at a source node that is at a distance d away from its destination, the clock will be initially positioned to align with the first bit of the $(d+1)$-th bit-group. If deflection does not occur, the clock in $\mathcal{H}$ will be shifted by one bit-group position to the right at each hop. When the clock is eventually shifted to the 0th bit-group, the packet is at its destination node and the node will attempt to absorb the packet. However, if a deflection should happen to the packet at a certain node, the clock is shifted by $(k-1)$ bit-groups to the left. This agrees with the fact that the deflection cost due to single deflection is k in the shuffle networks.

Since this self-routing scheme only requires the clock to be shifted in a fixed manner, it is relatively easy to implement. Regardless of the size of the network, routing decisions at each node are made by reading only three bits. Not only does this lead to a much lower latency in the header processing compared with searching a look-up table, the number of demultiplexers at each node remains constant independent of the size of the network. This makes it possible to design nodes in optical networks cost-effectively.

2.4 Routing Decision Making

In this section, we specify how to control the main routing switch shown in Fig. 3. The states of the four $LiNbO_3$ switches are configured by an electronic controller with a high-speed processor programmed to perform the routing decisions within the duration of a time slot. Operating in a time-slotted network allows the controller to be easily synchronized with the arrival of incoming packets at both inputs by using the frame pulse carried by each packet. This greatly reduces the complexity of the design of the controller compared with operating in an asynchronous mode. Also, packet synchronization leads to a much higher network throughput with deflection routing.

Assuming that all packets in the network are of equal priority, the objective of routing control is to achieve maximum network throughput while providing fairness in the treatment of packets arriving at the two input ports. According to Little's formula, the network throughput is inversely proportional to the average number of hops D that a packet takes between a random pair of source and destination nodes in steady state. To minimize D, the controller must minimize the number of deflections, especially deflections upon arrivals at destinations, and also minimize the delay due to buffering.

At any one of the network nodes, the controller examines the link requests at the inputs and assigns the output ports according to the following priority scheme:

1. *Packets Contending for the node receiver*: A packet in the memory loop destined for this node have priority and are absorbed first. If there is another packet at one of the input ports contending for the receiver, it is buffered. A third packet arriving for this node on the input is deflected to output port two.

2. *Packets contending for the same output*: A packet in the memory loop destined for one of the output ports has priority over packets at the input ports. Incoming packets contending for an output that is already in use are will be buffered if the memory is available or deflected if the memory is inaccessible due to blocking or usage by another packet. Contentions between packets at the two input ports are resolved by randomly selecting one of them to be buffered or deflected while the other is routed correctly.

3. *Don't care packets*: Since don't care packets can be routed to either one of the two output links, they are service after the requests of other packets.

4. *Packet generation*: A node transmitter has the lowest priority and only injects new packets into the TX port if it does not cause contention.

This ensures that at least one packet will be received when there are one or more packets need to be absorbed. A packet in the memory loop will always be assigned to its desired output port and can never be deflected.

2.5 Network Throughput

The steady-state behavior of the network with the specified routing schemes is investigated for an 8-node and a 64-node ShuffleNet. The offered traffic is assumed to be uniform throughout the network, i.e. the new packet generation probability g is the same at every node. Since no input buffer is provided, a newly generated packet will be either inserted if there is a slot available, or discarded otherwise. In steady-state, the aggregate network throughput $\mathcal{T}$ is defined as the average number of packets received per time slot in the network. In networks with no packet loss, this is also equal to the average number of packets injected per slot in equilibrium.

Due to the blocking nature of the main routing switch in our demonstration, packet loss can occur under certain circumstances. When the packet in the optical buffer takes output port 2, one of the two incoming packets will be dropped to the receiver (RX) undesirably if both packets wish to access the buffer and/or output port 1. In order to keep the network design simple and avoid packet regenerations, the packets are simply discarded and therefore lost.

The normalized curves for packets injected and received at different network load for 8-node and 64-node ShuffleNet are shown in Fig. 4a. The difference between the average number of injected and received packets corresponds to the number of packets lost due to blocking. Packet loss is rather insignificant when the network size is small, but increases steadily with the offered network load in the 64-node ShuffleNet and limits the throughput to 25%. Figure 4b shows that, with a nonblocking main routing

switch, the throughput in both the 8-node and 64-node networks are slightly higher.

The network delay due to deflections can be measured by the increase in the expected number of hops, D, that a packet visits between a random pair of source and destination nodes. According to Little's Formula,

$$2Nu = \mathcal{T}D \tag{4}$$

where N is the number of nodes in the network, and u is the time slot utilization. If we let r be the probability of a packet at one of the inputs destined for the node receiver, then the probability of a packet being received in a given time slot at each node, i.e. throughput per node, is

$$\mathcal{T}/N = 2ur \tag{5}$$

Equations 4 and 5 imply that $D = 1/r$, where r is calculated as $\mathcal{T}/2Nu$. The results are shown in Fig. 5.

Since the size of the network in our experiment was primarily limited by optical loss, the blocking routing switch provides sufficient advantages in loss performance to offset the degradation in throughput and network delay for networks with reasonable sizes .

3 Subsystem Experiments

A prototype network node is built to demonstrate experimentally the architecture established in section 2. The component stages involved in the system demonstration

are described here. This includes the optical packet generation, header recognition, the implementation of the self-routing scheme, controller electronics, and packet demultiplexing at the receiver.

3.1 Optical Packet Generation and Compression

Electronic packets are produced by a network control program running on a computer. For a pair of user-specified or randomly generated source and destination nodes, the program computes the appropriate packet header according to section 2.3. For testing purposes, one additional bit was included in each bit-group making the header a total of 16 bits and is set to 0 in our experiment. Packets generated with 16-bit packet header from the computer are sent to a pattern generator (HP71603B) which is externally synchronized with a 1.313 μm Nd:YLF 100MHz mode-locked laser. The outputs of the pattern generator are modulated with the optical pulse stream from the laser to produce optical packets at 100 Mbps. These packets are then optically compressed to 100 Gbps before being injected into the network. For simplicity, payload is set to all zeros in our demonstration, so the packets at the input of the compression stage are 16 bits long at a rate of 100 Mbps.

The configuration of the optical packet compressor is a feed-forward delay line structure as shown in Fig. 6. Each stage is a Mach-Zehnder lattice with one arm providing extra time delay of $T-\tau, 2(T-\tau), 4(T-\tau), .., 2^{n-1}(T-\tau)$ with respect to the other, where T is the incoming bit period and τ is the outgoing bit period. The number of bits in the packet is N and the number of stages required is $n = log_2(N)$.

At end of this structure, an $LiNbO_3$ modulator selects one of the compressed signals for transmission.

At the input of the compressor, the signal can be represented as

$$I_{in}(t) = \sum_{i=0}^{N-1} I_i(t) = \sum_{i=0}^{N-1} \delta(t - iT)A_i \tag{6}$$

where $I_i(t)$ is the ith bit in the packet and A_i is either a 0 or 1, depending on the bit pattern of the input packet. For a single optical pulse input, a group of N identical pulses with time delays of $T-\tau, 2(T-\tau), 3(T-\tau), .., (2^n-1)(T-\tau)$ with respect to the first one is composed after n stages. Therefore, the composed output for the ith input bit can be expressed by

$$O_i(t) = \frac{1}{2^{n+1}} \sum_{j=0}^{N-1} I_i(t - j(T-\tau)) \tag{7}$$

where the factor of $1/(2^{n+1})$ is the loss due to the signal splitting by the cascaded 2×2 couplers.

For the input packet of N bits, the output signal is described by summing O_i:

$$I_{out} = \sum_{i=0}^{N-1} O_i = \frac{1}{2^{n+1}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \delta(t - (i+j)T + j\tau)A_j \tag{8}$$

When $i+j = N-1$ is satisfied, consecutive N bits with the spacing of τ is compressed into the time frame of $(N-1)T$ with the packet build up time being $(N-1)T$.

A 4-stage structure consisting of five 2×2 couplers and a modulator was built for compressing 16-bit packets at 100 Gbits/sec. The splitting loss of the device is 22 dB.

3.2 Packet Header Demultiplexing

The central component necessary for ultrafast OTDM receivers is a high speed optical demultiplexer. The Terahertz Optical Asymmetric Demultiplexer (TOAD) has been proven to be an ideal candidate for this function. Illustrated in Fig. 7, the TOAD is based upon a fiber optical loop mirror with the addition of a nonlinear element, semiconductor optical amplifier (SOA), positioned asymmetrically in the loop and a 90:10 directional coupler to allow injection of clock pulse to control switching.

When the clock pulse is not present, the device works as a simple linear loop mirror. The *Data In* pulse injected into the input port of the device splits into two counterpropagating pulses (*Data CW* and *Data CCW*) at the 50:50 directional coupler. Since these pulses have relatively low intensity, they experience the same linear propagation through the SOA and around the loop of fiber and therefore experience the same phase shift. When they recombine at the 50:50 coupler, the pulses destructively interfere at the output port and reflect back to the input port. In the absence of the clock pulse, nothing is switched to the output port.

When the clock pulse is present, TOAD switching occurs. The SOA is positioned in the loop at a precise location, Δx, from the exact center of the loop. This position is determined by the size of the switching window desired and is computed as $2n\Delta x/c_0$ where n is the index of refraction of fiber (approximately 1.45) and c_0 is the speed of light in a vacuum. As shown in Fig. 7, when the SOA is positioned appropriately, *Data CW* is the first pulse to travel through the SOA. Next, the high intensity clock

pulse is injected into the SOA and causes a depletion of the amplifier carrier density which results in a change of the effective index of the medium. Finally, the *Data CCW* pulse travels through the SOA, but now it sees a different refractive index than *Data CW* and therefore experiences a different phase shift. If the clock pulse is set to an intensity such that this phase shift is π, the counterpropagating pulses interfere at the 50:50 coupler constructively and the data is is switched to the output port of the device. In order to distinguish data from clock at the output of the TOAD, it is necessary to inject them at orthogonal polarizations so that a linear polarizer placed at the output port can extinguish the clock and only allow the data to be transmitted.

The TOAD has been demonstrated with window sizes as small as 4 ps allowing it to demultiplex a single bit from a 250 Gbps bit stream. However, because of SOA carrier dynamics, the TOAD needs a small amount of time (~300-500 ps) to recover before demultiplexing another bit. Therefore, in order to be able to demultiplex a group of adjacent bits in an optical packet, it is necessary to use an array of these devices. A typical setup for demultiplexing a set of four bits is shown in Fig. 8. Four TOAD devices are illustrated with data and clock signals distributed to them. The lengths of the data and clock distribution fiber cables are chosen so that each TOAD demultiplexes a unique bit from a 4-bit group of the incoming data stream. This type of arrangement can be extended to an arbitrary number of bits and is only limited by the data and clock power available.

3.3 Packet Demultiplexing at Receiver

High bit rate packets must be received by each node in the network. For bit rates at and above 100 Gbps, standard optical receivers and electronics are inadequate. Instead, a fast all optical device must be used to recover the data in the packet payload. We propose using the TOAD device in two different configurations to receive the payload. The first configuration is a simple parallel scheme similar to that used to detect the packet header bits, (Fig. 9a). TOADs are constructed with a switching window, Δt, equal to the bit period, τ. The packet payload and clock are distributed to each TOAD via $1 \times M$ splitters. The clock lines for the TOADs have unique path lengths so that each TOAD can demultiplex a different bit within the payload. For a very small payload, 8 bits (1 byte), two 1×8 splitters, an array of 8 TOADs in parallel, and low bandwidth photodetectors could be used to receive the entire payload. However, for many application, it is desirable to support much larger payloads. Instead of using N TOADs where N is the number of bits in the payload, the parallel array can be used multiple times to receive a single payload. The number of TOADs required to build the array is determined by the recovery time of each TOAD, the bandwidth of the photodetector, and the speed of the digital thresholding electronics that receive the TOAD outputs. If we group the delay needed for the photodetector and electronics into a single term, t_{rec}, we find that the design parameter for the parallel configuration is given by t_p.

$$t_p = max(t_{TOAD}, t_{rec}) \tag{9}$$

where t_{TOAD} and t_{rec} represent the minimum recovery time for the TOAD and the speed of the receiver electronics respectively. The number of TOADs required for the parallel receiver, M, and the number of passes through the array, P, to demultiplex N bits of payload are given by $M = \lceil \frac{t_p}{\tau} \rceil$ and $P = \lceil \frac{N}{M} \rceil$

Unfortunately, the parallel configuration may involve an array of 16 or more TOADs to receive the packet payload. A more cost effective approach is to design a payload receiver using TOADs that scales logarithmically. This "serial" configuration of TOADs is shown in Fig. 9b. In this case, the goal of the receiver is to downconvert the high bit rate payload to a slow bit rate that can be handled by the receiver photodetector and thresholding electronics. The Mach-Zehnder version of the TOAD is used in an up/down toggle configuration. This permits the TOAD to output both the transmitted and reflected portions of the signal via "up" and "down" output ports [19]. The signal is switched "up" ("down") in the presence (absence) of the clock pulse. The design parameter for this system, t_s, is determined by

$$t_s = max(t_{TOAD}/2, t_{rec}) \tag{10}$$

The limiting speed is now only a function of half of the TOAD recovery time since the first TOAD actively switches every other incoming bit to the up port. Each TOAD is designed with a unique window, Δt, that will divide the incoming data in half between the up and down ports. The M^{th} TOAD splits the payload in half sending the first half (bits 0 through $\frac{N}{2}-1$) directly to the $(M-1)$ TOAD and delaying the second half (bits $\frac{N}{2}$ to $N-1$) by $2^{M-1}t_s$. The halving process continues with each TOAD using a

smaller window than its predecessor until the data has been downconverted to a rate that can be handled by the photodetector and digital thresholding electronics. The number of TOADs required for the serial receiver, M, grows logarithmically ($=log_2 N$) with the number of payload bits, N.

3.4 Pointer Control for Self-routing

The self-routing scheme described in section 2.3 is implemented using two crossbar switches as shown in Fig. 10, The polarizing beam splitter (PBS) splits the incoming optical packet into header, payload and clock. Demultiplexers in each node read one group from the header and send them to the routing controller. The routing controller determines the settings of the main 4×4 switch to route the packet appropriately. If the routing controller deflects a packet, the two crossbar switches are set in an exclusive manner so that either the clock is delayed or the packet itself is delayed with respect to the clock. Delaying the clock means that the clock is shifted by 4-bit positions backward to the previous group, and delaying the packet itself implies that the clock is shifted forward by $3(k-1)$-bit positions along the direction of packet propagation.

3.5 Controller Electronics

The electronic routing controller is designed to perform two main tasks: route packets and record the test packet transitions for the single-node network demonstration. Fig. 11 shows the schematic of the controller board, which is a dedicated piece of

hardware capable of handling all routing and one-node network simulation functions independently. It is interfaced to a computer through the ISA bus to allow the computer to inject new packets into the network and collect routing statistics upon the completion of each trial.

In the initial state, the computer randomly selects a pair of source and destination nodes and generates a new test packet with the appropriate routing tag. It then assigns this source node address to the controller to start the trial, and queues the test packet in a buffer on the controller board. Routing decisions are performed by a preprogrammed routing control chip, which is a high-performance CMOS EEPROM-based programmable logic device (EPLD) built on multiple array matrix architecture with 15ns pin-to-pin logic delay and counter frequency at 125 MHz. The controller board is synchronized with the packet clock, and checks the incoming traffic at the beginning of each time slot. If the desired output port is free, the test packet will be injected into the network. It takes approximately 20 ns for the controller to make a routing decision and produce the control signals to set the main routing switch and the deflection control switch to appropriate states. If any packet is sent into the optical buffer, the control chip will store the necessary information for routing in the next time slot.

Upon the departure of the test packet from the current node, the control chip computes the next node address in the network that the test packet will arrive at, and reassign the next node address to the current node address. This address calculation requires a computation time close to 100ns, which limits the packet rate in

the demonstration to be $\leq$ 7.7 MHz. All the routing actions performed on the test packets are saved in a 256 $\times$ 8 bits FIFO, to be read by the computer at the end of each trial.

Efficient design of routing electronics in transparent optical networks is critical since routing decisions should be made within the duration of one time slot. Networks with complex structure inevitably require relatively long processing delays, which reduce the packet rates that can be achieved. Parallel processing and pipeline techniques can be used with multi-processors designs to improve the performance.

3.6 Main Routing Switch

Four $LiNbO_3$ crossbar switches are connected together to form the switch in Fig. 3. Due to the critical timing accuracy necessary in OTDM networks, it was necessary to build the switch using precision fiber cutting and splicing techniques. The switch that was fabricated had an average input/output delay of 34 ns with a maximum error of less than 2 ps. The average loss from input to output was 7.5 dB, and the lithium niobate switches limited the maximum switching rate to 1 GHz.

4 Network Demonstration

The schematic of the experimental prototype of a single network node is shown in Fig. 12. To demonstrate the functionality of all of the network subsystems as well as the system as a whole, a set of experiments were designed and conducted to test each block. These tests were based upon a typical routing path through the

ShuffleNet network. To demonstrate a typical network routing scenario, a path was chosen originating at node 0 and terminating at node 7 as illustrated in Fig. 13. This requires the packet to travel through intermediate nodes, node 4 and node 1, of the network. Since only one node was constructed in our demonstrations, it was necessary to simulate the 8-node ShuffleNet by injecting packets into the node as though they had come from the previous node and allowing the node to dynamically adjust its current node address. This requires that the entire packet be regenerated before injection into each node.

To demonstrate routing functionality, the packets were generated at a slower bit rate of 400 Mbps. The packet format, illustrated in Fig. 14, has two components: the packet header and the packet payload. As described earlier, the packet header consists of 16 bits. The bit spacing is $\tau = 2.5ns$ and the packet header period is $t = 40ns$. The packet payload is set to all zeroes for the experiment. The length of each time slot in our system is $T = 160ns$.

For the experimental path from node 0 to Node 7, the packet header required is the following: 1000_0100_0100_0110. This header was generated using a 400 Mbps packet generation subsystem and injected into the network. As shown in Fig. 12, this packet was split into two components: a component for packet header demultiplexing and a component for network routing. To demonstrate the packet header demultiplexing subsystem, each of the 4-bit groups of the 16-bit header were demultiplexed at each node. Illustrated in Fig. 15a, demultiplexing of 3 of the 4 header bits is shown (the

fourth bit is always zero and therefore is not shown). The data indicates that the TOADs are properly configured and timed to demultiplex the appropriate bits of the header. Each vertical time line represents the demultiplexed group at a particular node in the packet routing path, (i.e. node 4 reads the second group, 0100, etc). Finally, the output of the header demultiplexing subsystem after thresholding and conversion to ECL digital logic levels is shown in Fig. 15b.

The outputs of the header demultiplexing subsystem are then connected to the routing control board. The functional operation of the routing control board switch outputs for the routing path described above is shown in Fig. 16a. In this diagram, a low signal corresponds to setting one of the crossbar switches to the bar state, and a high signal corresponds to the cross state. Finally, the outputs of the 4×4 routing control switch, shown in Fig. 16b, demonstrate the correct routing of the packets through the network. This can be seen by considering the path that the packet travels through the network. In the first time slot, the packet exits node 0 via output port 1. In the second and third time slots, the packet exits output port 2 of node 4 and node 1, respectively. In the fourth time slot, the packet is received through the RX port at node 7.

5 Conclusions

This report has presented the design and experimental demonstration of a transparent optical TDM packet-switched network. The particular network architecture,

ShuffleNet, was chosen because of its symmetric properties and ease of scalability. Using the self-routing feature in this network, combined with deflection routing as the contention resolution principle, low processing delay at each node in the network has been achieved. With the components developed for operating the network at ultrafast bit rates, we have successfully demonstrated packet routing through designated paths in a simulated 8-node shuffle network.

References

[1] A. S. Acampora, M. J. Karol, "An overview of lightwave packet networks," *IEEE Network Magazine*, pp. 29-40, Jan. (1989).

[2] B. Mukherjee, "WDM-based local lightwave networks, Part I: Single-hop systems," *IEEE Network Magazine*, vol 6, No. 3, pp. 12-27, May, 1992.

[3] B. Mukherjee, "WDM-based local lightwave networks, Part II: Multihop systems," *IEEE Network Magazine*, pp. 20-32, July 1992.

[4] D. M. Spirit, A. D. Ellis, P. E. Barnsley, "Optical time division multiplexing: systems and networks," *IEEE Communications Magazine*, Vol 32, n 12, pp. 56-62, Dec. 1994.

[5] R. A. Barry, et. al., "All-optical network consortium-ultrafast TDM networks," *IEEE Journal on Selected Areas in communications*, vol 14, pp. 999-1011, June 1996.

[6] P. R. Prucnal, M. A. Santoro, S. K Sehgal, and I. P. Kaminow, "TDMA fiber optic network with optical processing," *Electronics Letters*, vol. 22, pp. 1218-1219, 1986.

[7] M. G. Kane, I. Glesk, J. P. Sokoloff, P. R. Prucnal, "Asymmetric optical loop mirror analysis of an all-optical switch," *Applied Optics*, vol. 33, pp. 6833-6842, 1994.

[8] J. P. Sokoloff, P. R. Prucnal, I. Glesk, M. Kane, "A terahertz optical asymmetric demultiplexer (TOAD)," *IEEE Photonics Technology Letters*, vol. 5, no. 7, pp. 787-790, 1993

[9] I. Glesk, J. P. Sokoloff, P. R. Prucnal, "Demonstration of all-optical demultiplexing of TDM data at 250 Gbit/s," *Electronics Letters*, vol. 30, pp. 339-340, 1994.

[10] I. Glesk, J. P. Sokoloff, P. R. Prucnal, "All-optical address recognition and self-routing in a 250 Gbit/s packet-switched network," *Electronics Letters*, vol. 30, pp. 1322-1323, 1994.

[11] S. W. Seo, K. Bergman, P. R. Prucnal, "Transparent optical networks with time-division multiplexing," *IEEE Journal on Selected Areas in Communications*, vol 15, June, 1996.

[12] E. Ayanoglu, "Signal flow graph for path enumeration and deflection routing analysis in multihop networks," *IEEE Trans. Commun.*, vol. COM-40, pp. 1082-1090, June 1992

[13] K. Padmanabhan and A. N. Netravali, "Dilated networks for photonic switching," *IEEE Trans. Commun.*, vol. COM-35, pp 1357-1365, Dec. 1987

[14] M.G. Hluchyj and M.J. Karol, "ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks," in *Proc. of IEEE INFOCOM,* R. Rutledge and A. Leon-Garcia, eds.(IEEE, New York, 1988), pp. 379-390.

[15] N.F. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks," in *Proc. of IEEE INFOCOM*, C. Desmond and J. Silvester, eds.(IEEE, Washington D.C., 1989), pp. 800-809.

[16] D.H. Lawrie, "Access and alignment of data in an array processor," *IEEE Trans. Comput.*, Vol.C-24, pp. 1145-1155, Dec. 1975.

[17] D.M. Spirit, A. D. Ellis, and P.E. Barnsley, "Optical time division multiplexing: Systems and networks," *IEEE Communications Magazine*, pp. 56-62, Dec. 1994.

[18] P.R. Prucnal, "Optically-process self-routing, synchronization and contention resolution for 1D and 2D photonic switching architectures, *IEEE J. Quantum Electronics*, 29, Vol. 2, pp. 600-612, 1993.

[19] K. I. Kang, I. Glesk, T. G. Chang, P. R. Prucnal, and R. K. Boncek, "Demonstration of all optical Mach-Zehnder demultiplexer," *Electronic Letters*, vol. 31, no. 9, pp. 749-750, 1995.

[20] P. A. Perrier and P.R. Prucnal, "Self-clocked optical control of a self-routed photonic switch," *IEEE J. Lightwave Tech.*, LT-7, Vol. 6, pp. 983-989, 1986.

[21] A. Acampora and S. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Transactions on Communications,* Vol. 40, No.6, pp. 1082-1090, June 1992.

[22] A. Krishna and B. Hajek, "Performance of shuffle-like switching networks with deflection," *IEEE INFOCOM*, M. Desmond and J. Silvester, eds.(IEEE, Washington D.C., 1990), Vol. 2, pp.473-480.

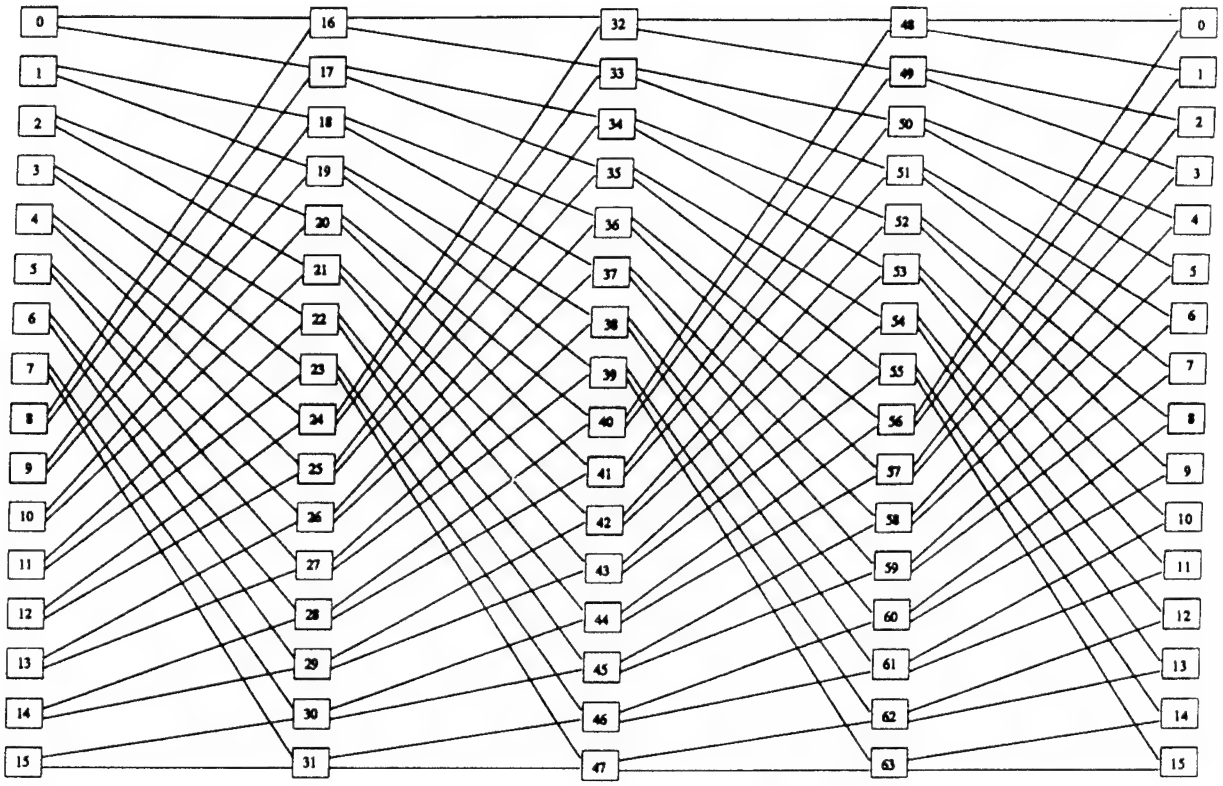


Fig. 1

Applied Optics, B. Yu

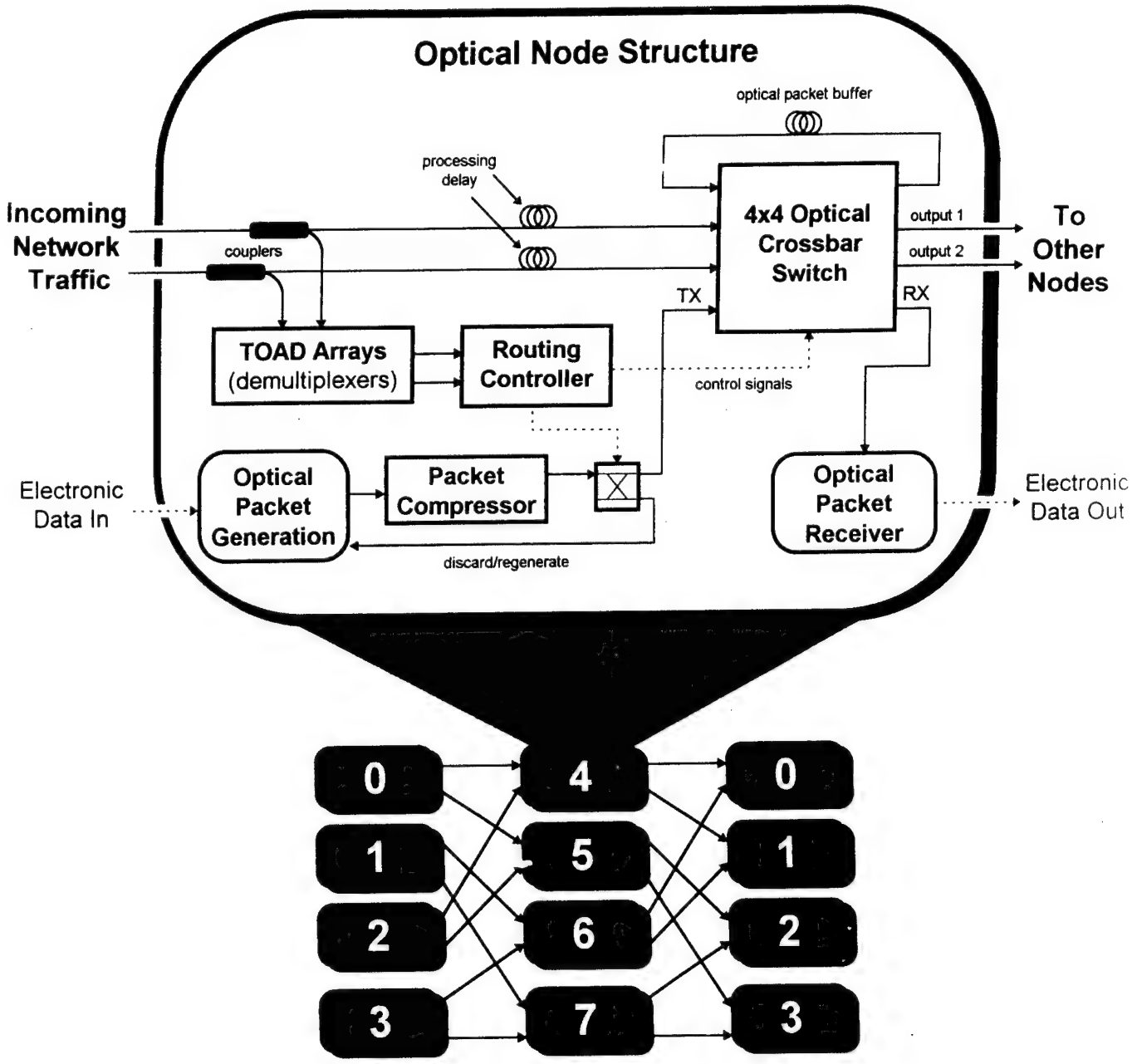

Optical Node Structure
optical packet buffer
processing delay
Incoming Network Traffic
couplers
4x4 Optical Crossbar Switch
output 1
output 2
To Other Nodes
TX
RX
TOAD Arrays (demultiplexers)
Routing Controller
control signals
Electronic Data In
Optical Packet Generation
Packet Compressor
Optical Packet Receiver
Electronic Data Out
discard/regenerate
0
1
2
3
4
5
6
7
0
1
2
3

Figure 2

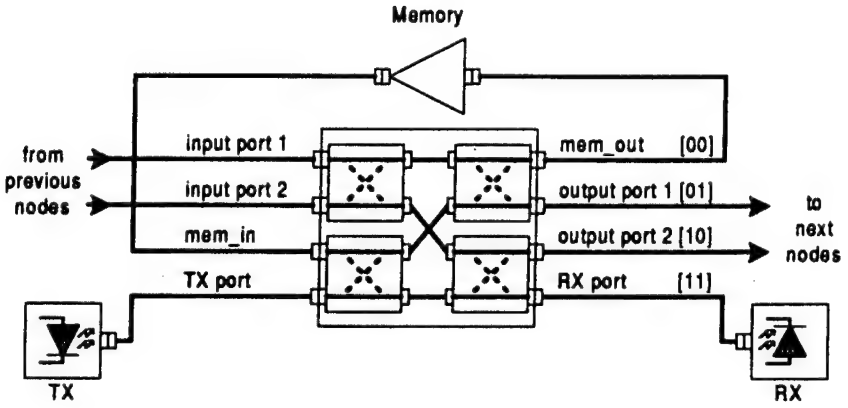


Fig. 3
Applied Optics, B. Yu

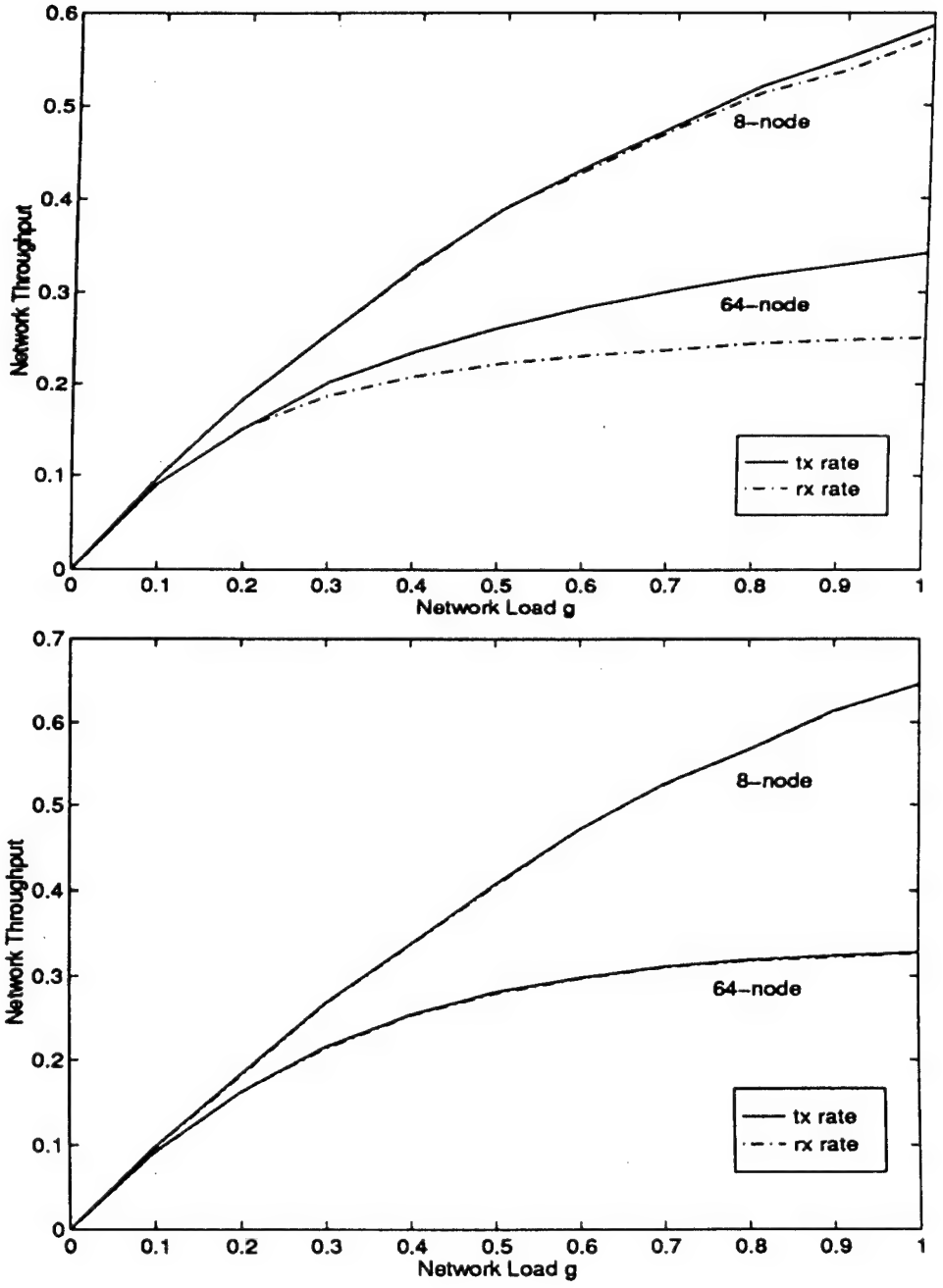


Fig. 4 (a),(b)
Applied Optics, B. Yu

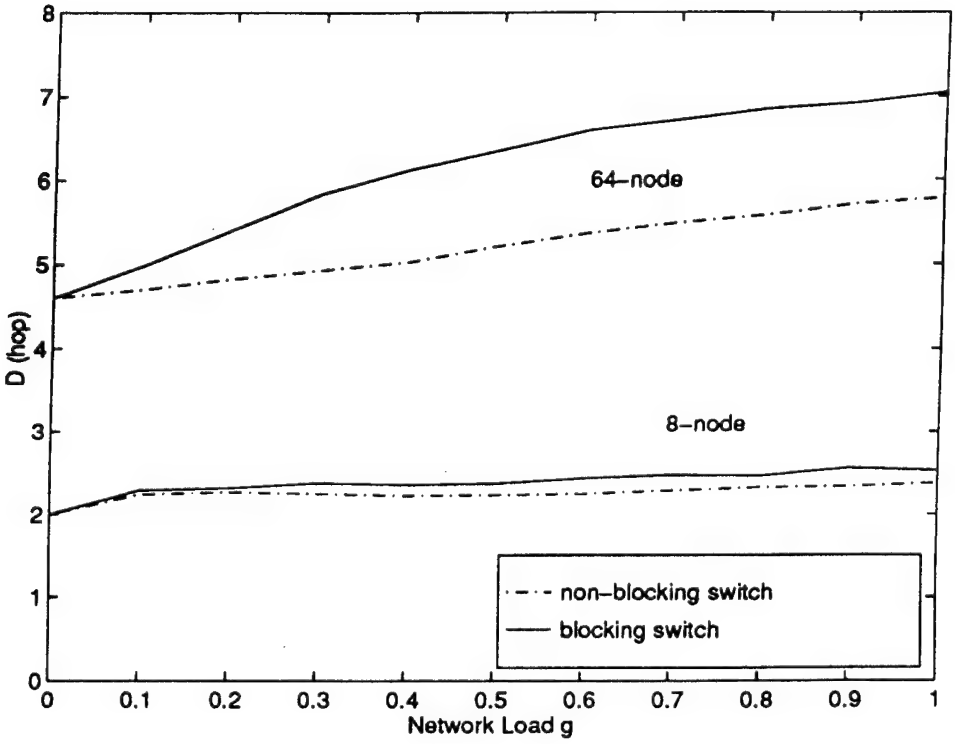


Fig. 5
Applied Optics, B. Yu

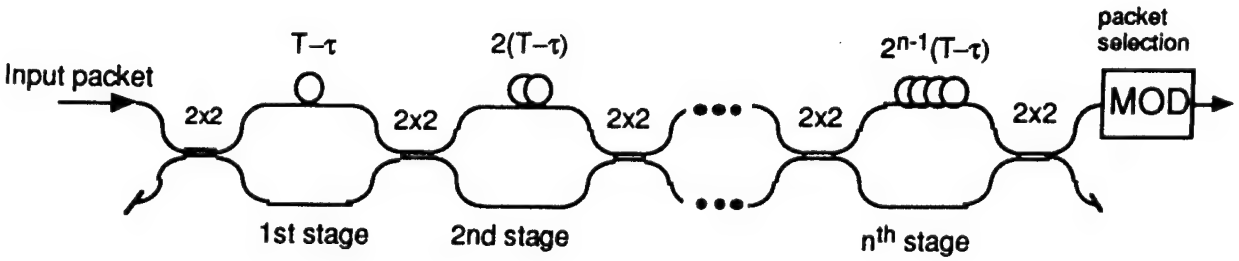


Fig. 6
Applied Optics, B. Yu

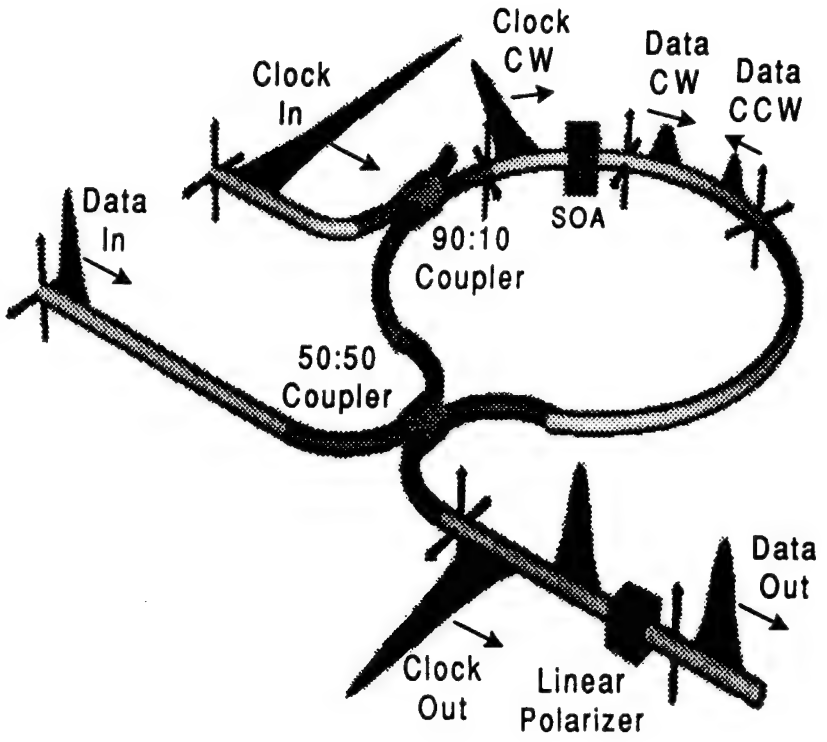


Fig. 7
Applied Optics, B. Yu

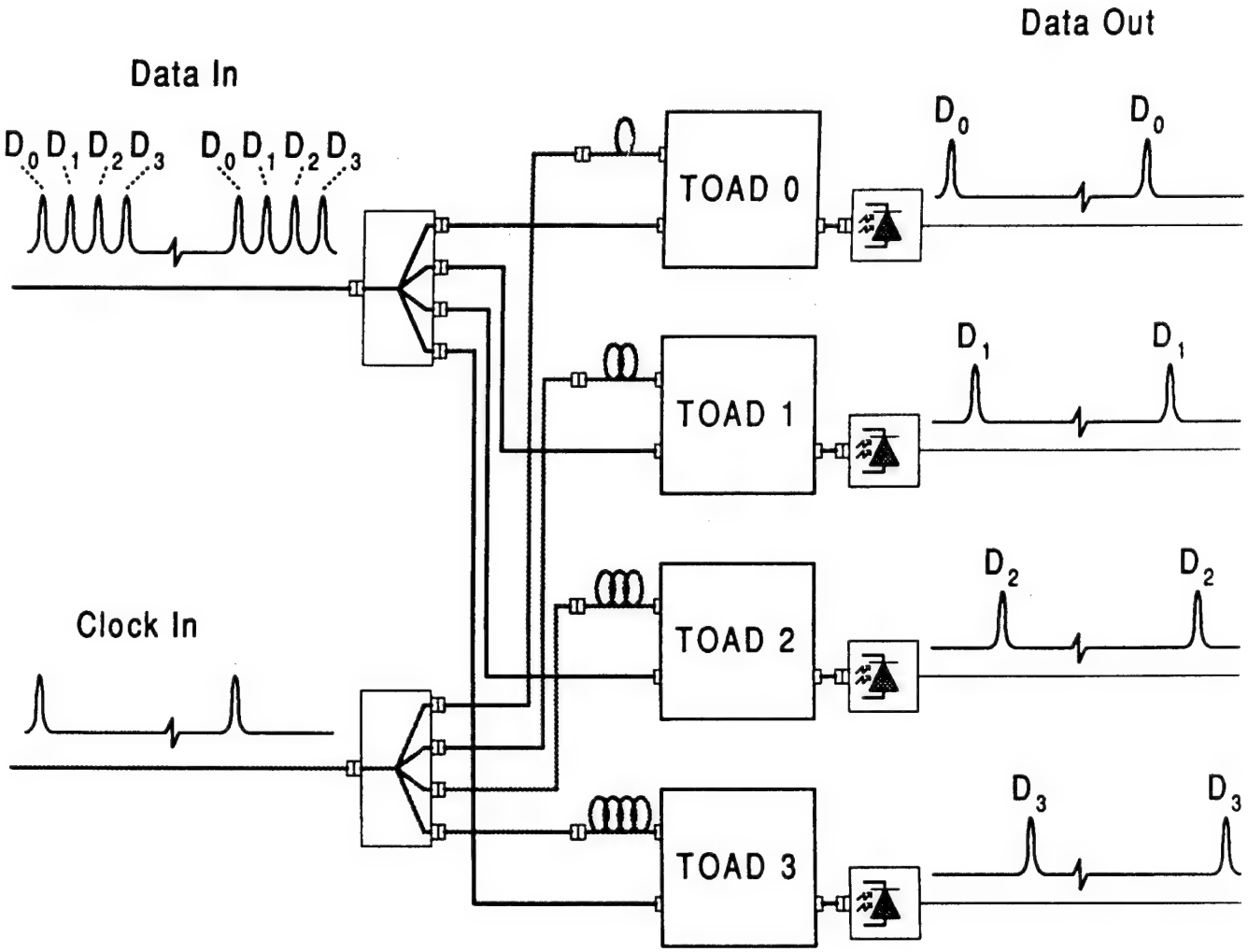
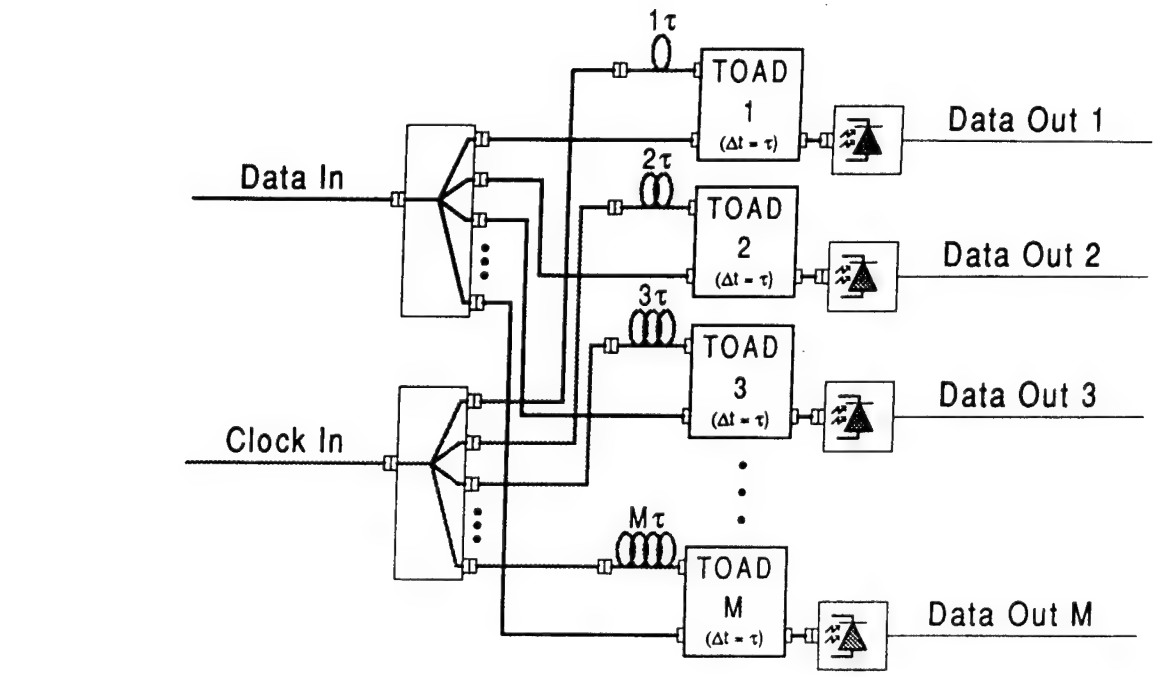
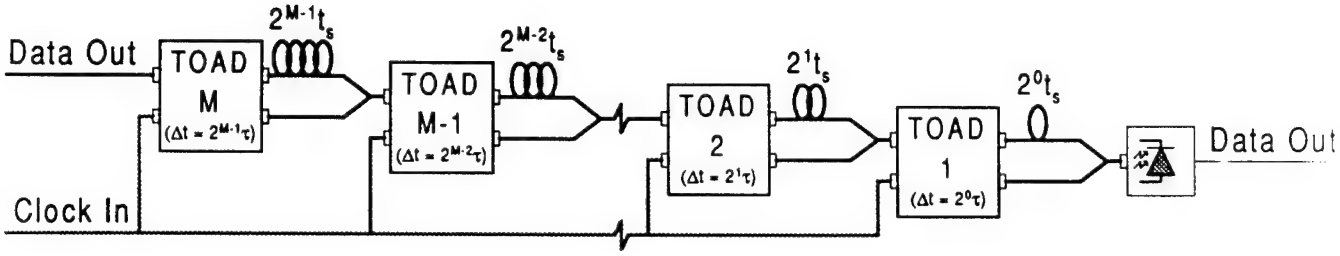


Fig. 8
Applied Optics, B. Yu



(a)



(b)

Fig. 9
Applied Optics, B. Yu

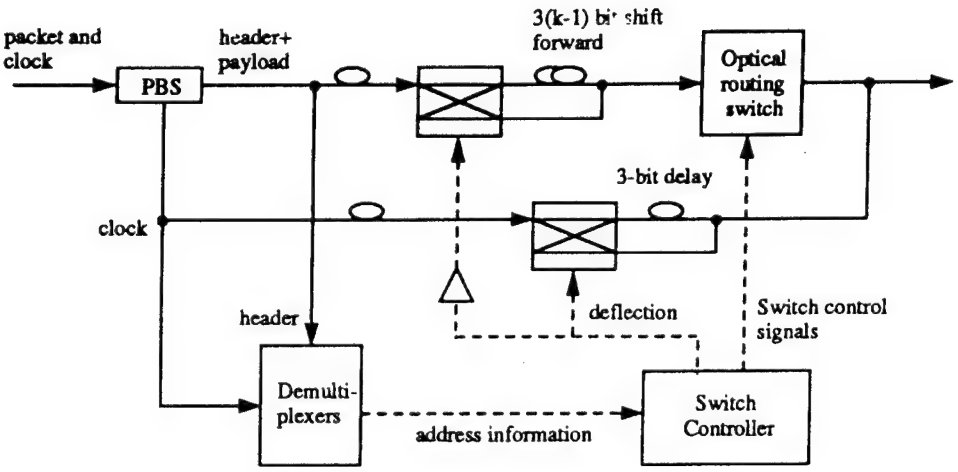


Fig. 10
Applied Optics, B. Yu

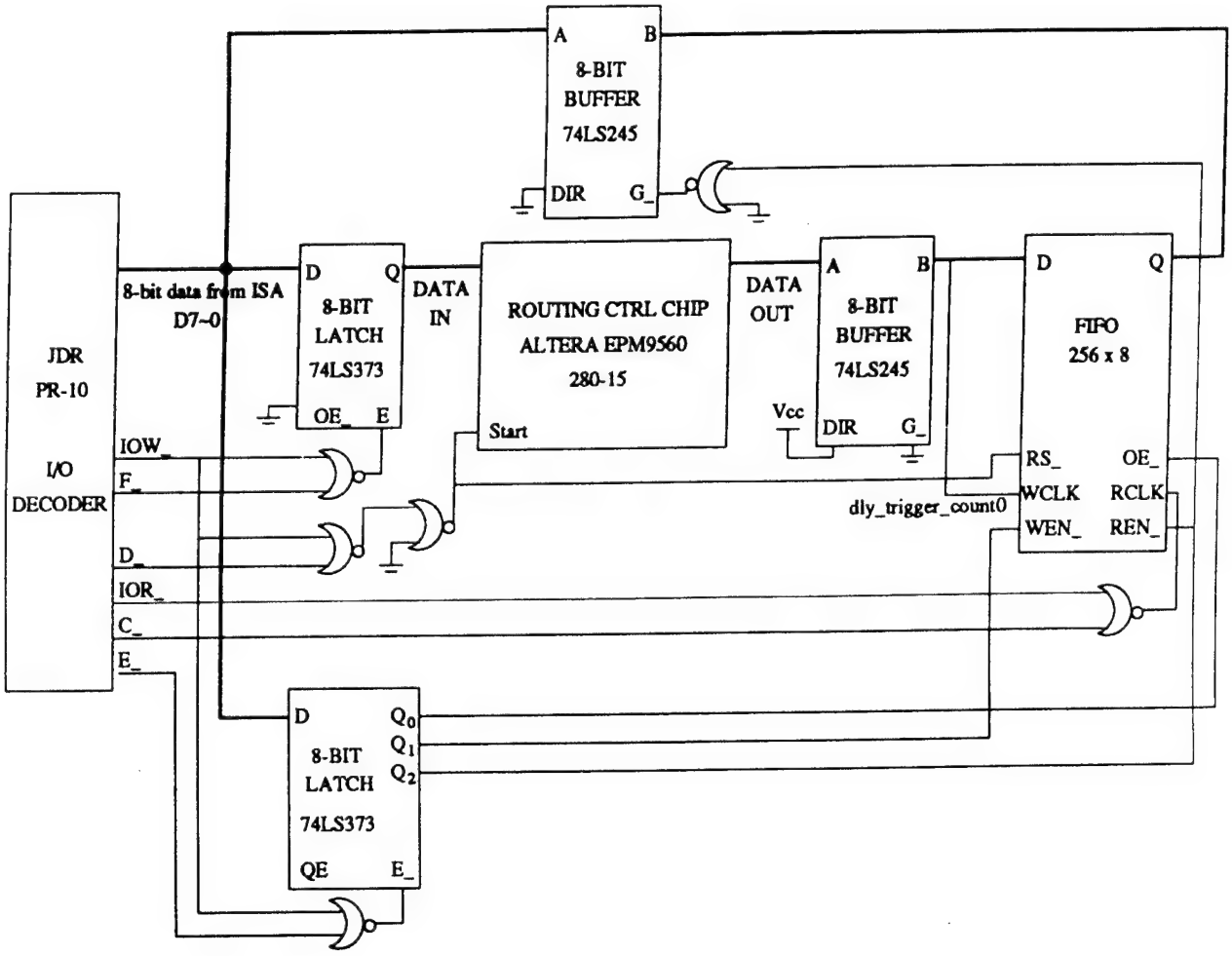


Fig. 11
Applied Optics, B. Yu

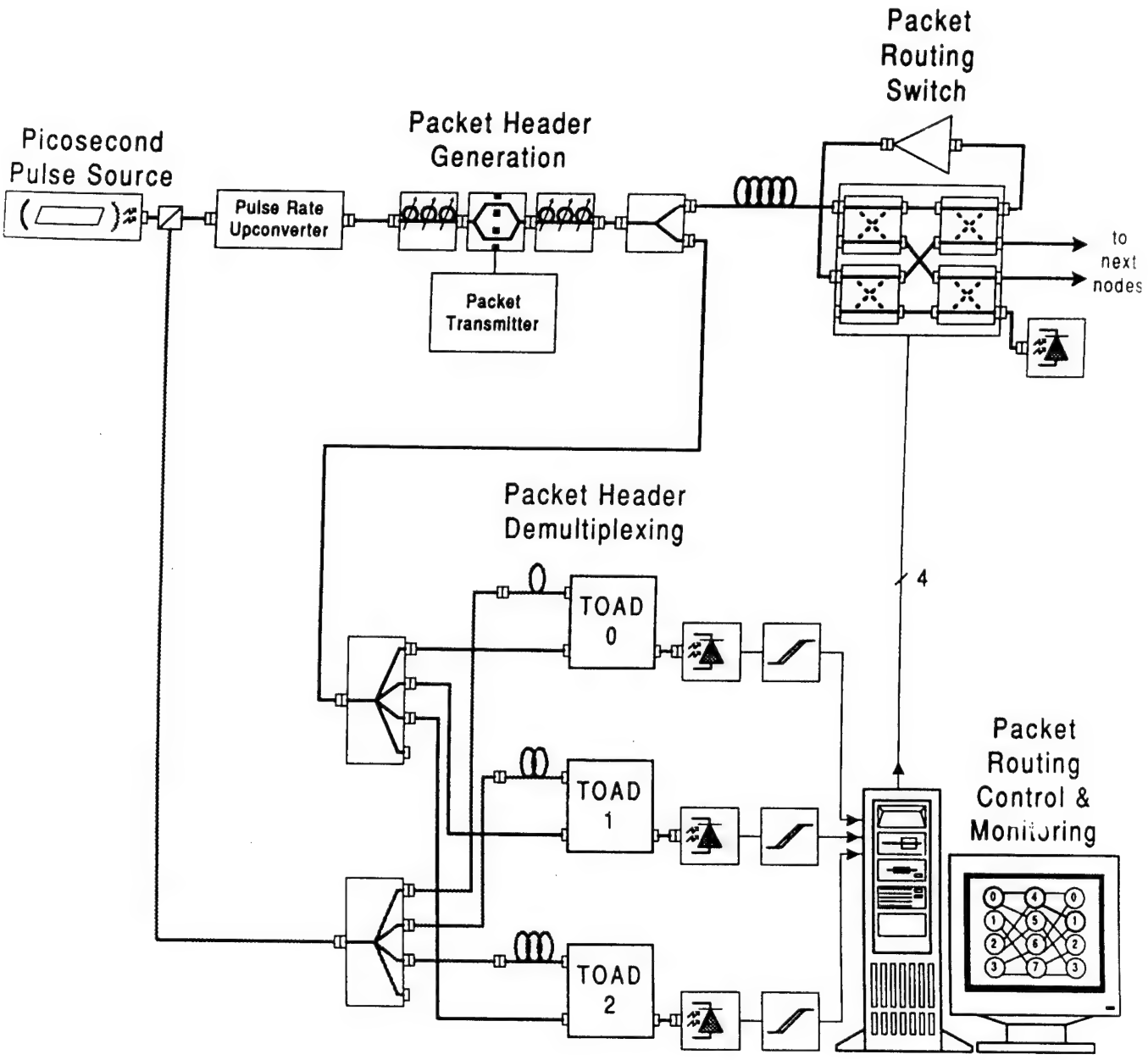


Fig. 12
Applied Optics, B. Yu

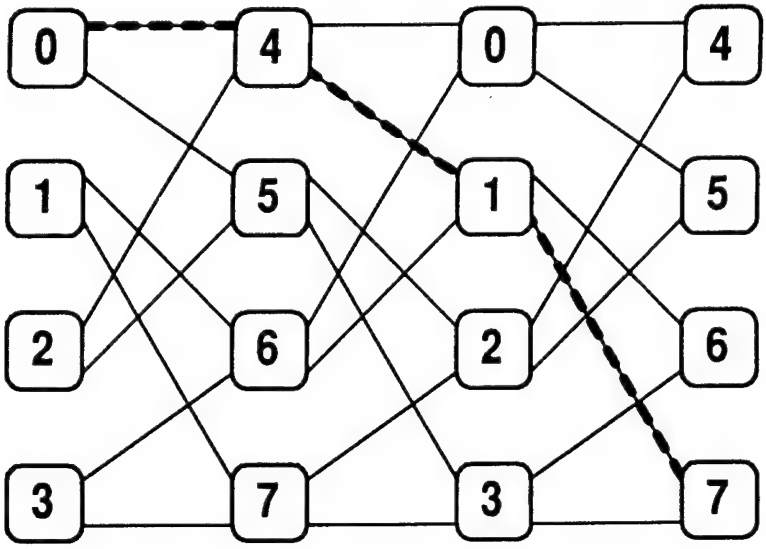


Fig. 13
Applied Optics, B. Yu

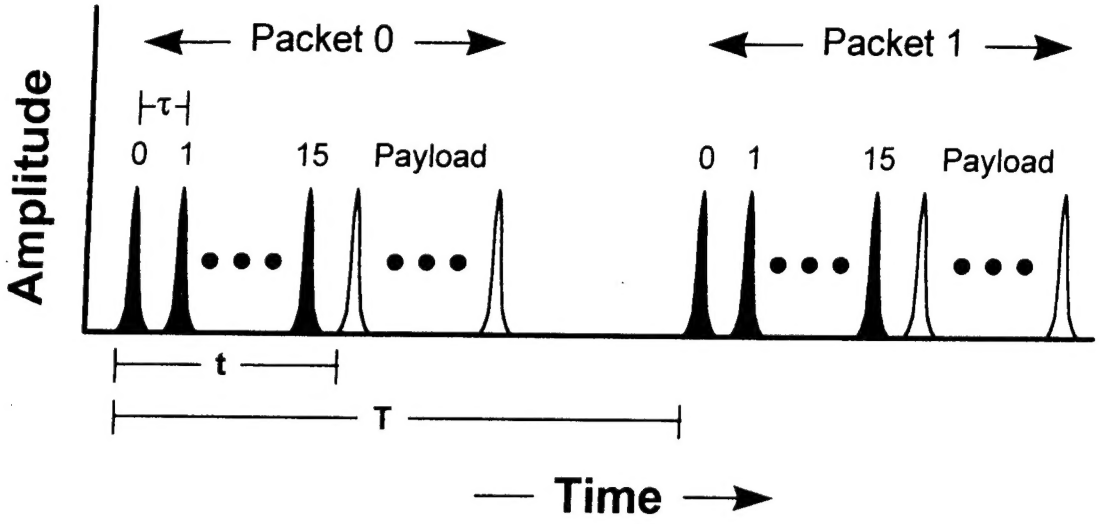
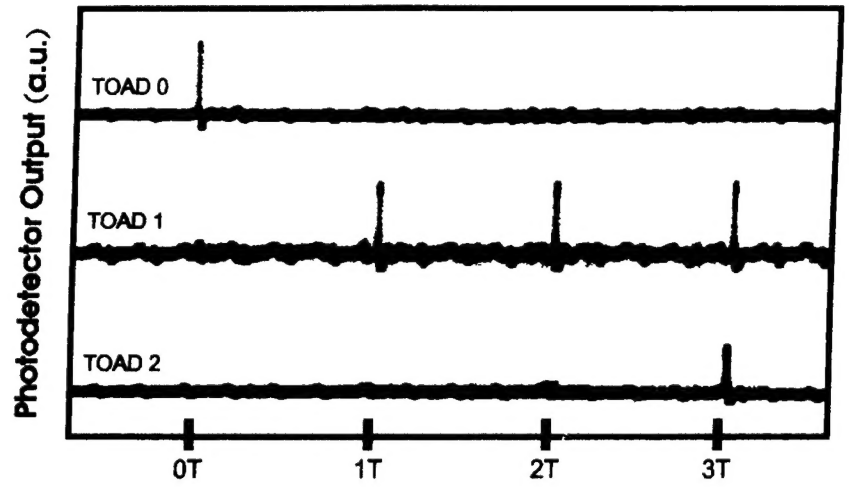
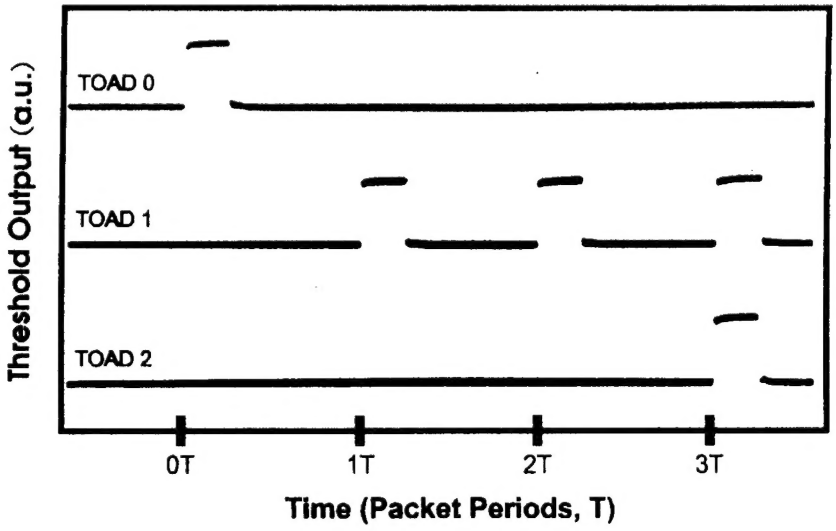


Figure 14

Applied Optics, B. Yu



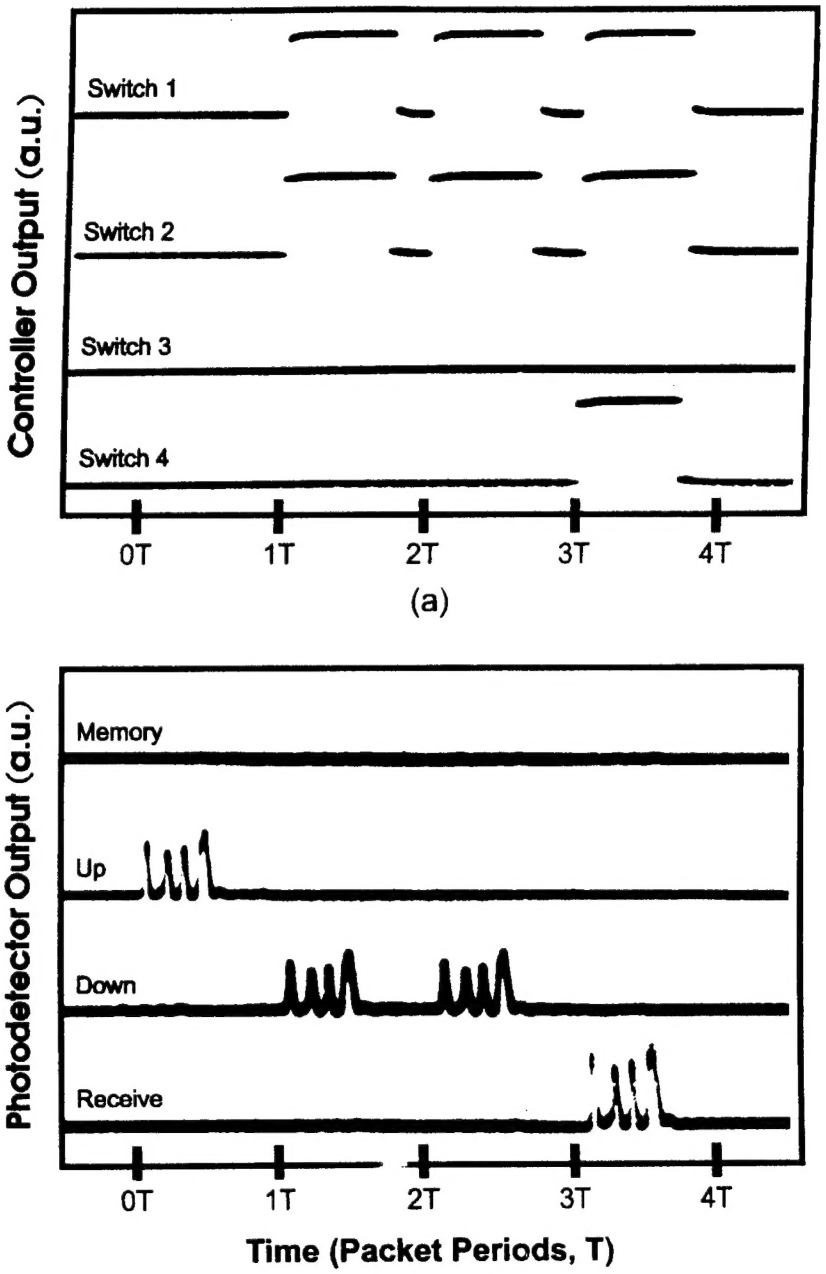
(a)



(b)

Figure 15

Applied Optics, B. Yu



*U.S. GOVERNMENT PRINTING OFFICE: 1998-610-130-61116

Figure 16

Applied Optics, B. Yu

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.